

H14/B03 プログラム変換による科学的ソフトウェア 作成法の研究(共同プロジェクト研究)

雑誌名	東北大学電気通信研究所研究活動報告
巻	9
ページ	195-197
発行年	2003-07
URL	http://hdl.handle.net/10097/30371

課題番号 H14/B03

プログラム変換による科学的 ソフトウェア作成法の研究

[1] 組織

代表者：高野 明彦（国立情報学研究所）
 責任者：外山 芳人（東北大学電気通信研究所）
 分担者：大堀 淳（北陸先端大学院大学情報科学研究科）
 亀山 幸義（筑波大学電子情報工学系）
 胡 振江（東京大学大学院情報理工学系研究科）
 長谷川 立（東京大学大学院数理科学研究科）
 小川 瑞史（科学技術振興事業団・さがけ研究21研究員、東大大学院情報理工学）
 細谷 晴夫（京都大学数理解析研究所）

研究費：校費 21万4千円
 旅費 28万7千円

[2] 研究経過

科学的なソフトウェア作成法の確立には、正しさが数学的厳密さで保証された高信頼なプログラム部品の蓄積と、それらを組み合わせて所望の機能と性能を備えるソフトウェアを構築する方法の開発が必須である。この観点から、多相型理論や論理学的基礎づけを持つ高信頼なプログラミング言語が提案され、データ構造の代数的性質を利用してプログラム部品を融合するプログラム変換法が研究されている。本プロジェクトでは、応用上ますます重要となったXMLを操作対象とするソフトウェアへの適用を視野に入れながら、プログラム変換やデータ変換を用いた科学的ソフトウェア作成法の探求を目指して研究を行った。

本プロジェクトは、本年度が初年度であった。参加研究者の主要研究分野である、高信頼プログラミング言語、プログラム検証、プログラム変換、データ変換等の最新の研究成果を踏まえて、新しい科学的ソフトウェア作成法の必要要件について議論することから研究を開始した。

以下、研究活動状況の概要を記す。

本年度は、年度前半に、国際会議等で参加研究者の多くが顔を合わせる機会が数回あったため、実質的な議論はそれらの場を借りて先行して行っ

た。そこでの議論を踏まえて、本プロジェクトとして第1回目の研究集会を11月6～7日に電気通信研究所で開催した。ここでは、参加者全員が各自の最新の研究成果について発表を行い、それらに基づく新しいソフトウェア作成法の可能性、XMLを操作対象とするソフトウェアへの適用可能性について研究討論を行った。

具体的な研究発表と討論の内容は次の通り。

外山(東北大)は、項書き換え系の完備化とプログラムの融合変換との興味深い関係について、最新の実験結果を含めて発表した。主に理論的な興味から深く研究されてきた完備化が、実行効率の改善を目的とするプログラム変換と同等の効率化を実現することの意味について議論した。

胡(東大)は、プログラムの仕様から並列プログラムを系統的に導出するプログラム演算法を紹介した。そこで重要な役割を果たす高階パターンマッチは一般には非決定的だが、何らかの制限を加えることによって決定的にできないかとの問題提起を行った。実際のプログラム仕様が満足する適当な制約条件が見つかれば実用上の価値は大きい。

亀山(筑波大)は、delimited continuationの議論で使われている shift・reset操作に健全で完全な公理化を与える方法について発表した。究極の制御構造ともいえるcontinuationはCPS-translationを通じてプログラム解析やコンパイラでの等価変換の基礎づけとなっており、実プログラムの意味に近い delimited continuationの公理化の意義は大きい。

長谷川(東大)は「Linear Logicは一般人に使えるか？」と題して発表した。高階ラムダ演算法(System F)に再帰コンビネータを加えると矛盾するが、高階線形論理に再帰コンビネータを加えても無矛盾である。このことから、「高階線形論理+再帰コンビネータ」に基礎をおくプログラミング言語を設計すべきとの議論になった。

大堀(北陸先端大)は、コンパイラのレジスタ割付処理が、ある種の証明変換として定式化ができることを示し、それに基づいて実装したプロトタイプ処理系を紹介した。中間言語からレジスタ数に制限のある機械語へ変換するレジスタ割付は、もっとも実マシンに近いプログラム変換と考えられるが、現在アドホックな手法が主流のこの分野で、

証明変換という高信頼な解決法は新たな定番となる可能性がある。

小川(JST/東大)は、bounded tree-widthグラフの代数的構成法を手がかりに、フローチャート言語のdead code検出等のプログラム解析が線形で行えることを示した。コントロールフローグラフの代数的構造を陽に表すSP項を導入し、プログラム解析の仕様をSP項上のcatamorphismとして記述することにより、maximum weightsum問題の最適化定理が適用可能となり、効率のよい線形アルゴリズムを得ている。組合せ理論の深い結果を実用的アルゴリズムの導出に結びつけている。

細谷(京大)は、XMLデータの静的な型チェックを行うプログラミング言語XDuceについて発表した。XDuceは、正規表現を用いてXMLデータの型定義が可能で、XMLデータに対する操作をより直截に記述できる。現在、C、Java、Haskellなどの既存言語のデータ構造にXMLデータを読み込んで処理する方法が主流だが、読み込み時のデータ構造変換が原因で、XMLデータの構造とは異なる構造を持つプログラムを書かねばならない場合も多く、信頼性や運用性の低下を招いている。

高野(情報学研)は、大規模なテキストデータについて、類似性評価などの連想計算を高速に行う汎用連想計算エンジンGETAについて発表した。GETAは16ノード規模のPCクラスタを用いて、1000万件規模のテキストDBを対象とする連想計算を高速処理できる。GETAの応用例として、国立情報学研が提供する図書情報検索サービス Webcat Plusがデモをまじえて紹介された。GETAで高速処理可能な計算について代数的モデルを構築し、プログラム演算によってインデックス分割を行う可能性について議論した。

以上8件の研究発表と討論を通じて、理論研究に裏付けられた新しいプログラミングへの模索が、見通しのよい成果を上げつつあることが確認できた。これらの進展を適切に取り入れて、新しい科学的なソフトウェア作成法の実現に向けて、引き続き研究を進める方向で討論を行った。

[3] 成果

(3-1) 研究成果

研究集会で次のような知見が得られた。

- (1) プログラム融合変換は、データの受け渡しに使われる中間データ構造を系統的に削除する手法として研究されてきた。変換手法は、ローカルな書き換えを繰り返す形で定式化されているが、その書き換えの戦略についてはほとんど研究されていない。外山の考察より、

完備化の手続きなどが、より適切な戦略を与えている可能性があることが分かった。

- (2) プログラム演算により、多様なプログラムの並列化を系統的に行えるが、その過程では人手による適切なガイドが必須となっている。胡の考察によれば、そのほとんどは人間の深い洞察を使用しておらず、自動化可能で、必要なことはどのような変換の可能性だけを対象とすべきか明らかにすることである。
- (3) 通常のcontinuationが、ある時点以降のすべての計算を表しているのに対し、delimited continuationは、ある時点以降の部分的な計算を表す。これは実際のプログラムの意味に近いと考えられ、この公理化により、従来よりも強力なプログラム変換法が定式化できる可能性が生まれた。
- (4) 高階線形論理と再帰機構は相性がよく、これをベースに、新しいプログラミング言語を設計すべきである。その際、線形論理のプリミティブに対応する言語の機構をどのように設計するかがカギとなる。それらは、通常のプログラマが直感的に使いこなせるものでなければならない。
- (5) レジスタ割付処理という、コンパイラの最もハードウェアに近い処理までが、証明変換という見通しのよい方法で解決できた意義は大きい。これで、対象言語の仕様とマシン語の仕様を与えて、コンパイラ全体を証明変換として系統的に構築する可能性が見えてきた。
- (6) 組合せ理論はグラフの探索などについて深い理論的結果を得ているが、それらはこれまで実プログラミングとは結びついていなかった。これは、線形アルゴリズムの存在が理論的に保証されても、実際、理論から導出されるアルゴリズムは巨大な定数項を持ち、実用上は無価値であることがほとんどだからである。小川らの研究により、実プログラムの解析対象となるグラフについては、この定数項を系統的に小さくできる方法が与えられ、理論と実践のギャップを埋める手がかりが得られた。
- (7) XMLデータの処理は、今後のプログラミングでますます重要になると考えられるが、現在のプログラミング言語が提供する解法は理想からほど遠い。XMLデータを仕様通りに取り扱って、プログラムの静的型チェックや各種プログラム解析が可能となる枠組みが求められている。XDuceでの取り扱いが合理的であり、XMLの実アプリケーションで使用可能な処理系のリリースが待たれる。

- (8) GETAの実用性はWebcat Plus等で実証されたが、現在の利用環境はCとPerlに限られており、普及を妨げとなっている。HaskellやJavaからも利用可能となることが期待される。また、GETAで高速処理可能な計算のクラスを明らかにして、それに対する代数モデルを構築すれば、GETAの連想計算にプログラム算法が適用可能となる。プログラム演算による実用的なソフトウェア開発の実例となる可能性が高い。

(3-2) 波及効果と発展性など

本プロジェクトは本年度から始まったものであり、本プロジェクトから生まれた研究成果と呼べるものは、まだ形になっていない。しかし、普段、学会等で出会っても議論を深める機会の少ない関連分野の研究者が、一堂に会してじっくり議論できたことで、参加者は大きな刺激を受けた。いずれ近い将来、本プロジェクトが契機となり、本プロジェクトの目標である科学的なソフトウェア作成法の確立に向けて、新しい研究が進展すると期待できる。

[4] 成果資料

- (1) Atsushi Ohori, Register Allocation by Proof Transformation, ESOP 2003, LNCS 2618, pp.399-413, 2003.
- (2) Zhenjiang Hu, Hideya Iwasaki, Masato Takeichi, An Accumulative Parallel Skeleton for All, ESOP 2002, LNCS 2305, pp.83-97, 2002.
- (3) Mizuhito Ogawa, Zhenjiang Hu, Isao Sasano, Masato Takeichi, Algebraic construction of graphs with bounded tree width and its applications - Catamorphic Approach to Program Analyses, APLAS 2002, pp.58-73, 2002.
- (4) Haruo Hosoya and Makoto Murata. Validation and boolean operations for attribute-element constraints. In Programming Languages Technologies for XML (PLAN-X), pages 1-10, 2002.
- (5) Akihiko Takano, Shingo Nishioka, Osamu Imaichi, Yoshiki Niwa, Association Computation for Information Access, APLAS 2002, pp.334-336, 2002.